



## RESEARCH ARTICLE

# Breeding novel solutions in the brain: a model of Darwinian neurodynamics [version 1; referees: awaiting peer review]

András Szilágyi<sup>1,3</sup>, István Zachar<sup>2,3</sup>, Anna Fedor<sup>1,3</sup>, Harold P. de Vladar<sup>3</sup>,  
Eörs Szathmáry<sup>1-4</sup>

<sup>1</sup>MTA-ELTE Theoretical Biology and Evolutionary Ecology Research Group, Budapest, H-1117, Hungary

<sup>2</sup>Department of Plant Systematics, Ecology and Theoretical Biology, Institute of Biology, Eötvös University, Budapest, H-1117, Hungary

<sup>3</sup>Parmenides Center for the Conceptual Foundations of Science, Munich/Pullach, 82049, Germany

<sup>4</sup>Institute of Advanced Studies, Kőszeg, H-9730, Hungary

**v1** First published: 28 Sep 2016, 5:2416 (doi: [10.12688/f1000research.9630.1](https://doi.org/10.12688/f1000research.9630.1))  
Latest published: 28 Sep 2016, 5:2416 (doi: [10.12688/f1000research.9630.1](https://doi.org/10.12688/f1000research.9630.1))

## Abstract

**Background:** The fact that surplus connections and neurons are pruned during development is well established. We complement this selectionist picture by a proof-of-principle model of evolutionary search in the brain, that accounts for new variations in theory space. We present a model for Darwinian evolutionary search for candidate solutions in the brain.

**Methods:** We combine known components of the brain – recurrent neural networks (acting as attractors), the action selection loop and implicit working memory – to provide the appropriate Darwinian architecture. We employ a population of attractor networks with palimpsest memory. The action selection loop is employed with winners-share-all dynamics to select for candidate solutions that are transiently stored in implicit working memory.

**Results:** We document two processes: selection of stored solutions and evolutionary search for novel solutions. During the replication of candidate solutions attractor networks occasionally produce recombinant patterns, increasing variation on which selection can act. Combinatorial search acts on multiplying units (activity patterns) with hereditary variation and novel variants appear due to (i) noisy recall of patterns from the attractor networks, (ii) noise during transmission of candidate solutions as messages between networks, and, (iii) spontaneously generated, untrained patterns in spurious attractors.

**Conclusions:** Attractor dynamics of recurrent neural networks can be used to model Darwinian search. The proposed architecture can be used for fast search among stored solutions (by selection) and for evolutionary search when novel candidate solutions are generated in successive iterations. Since all the suggested components are present in advanced nervous systems, we hypothesize that the brain could implement a truly evolutionary combinatorial search system, capable of generating novel variants.

## Open Peer Review

**Referee Status:** Awaiting Peer

Review

**Discuss this article**

Comments (0)

**Corresponding author:** András Szilágyi ([and.szilagyi@gmail.com](mailto:and.szilagyi@gmail.com))

**How to cite this article:** Szilágyi A, Zachar I, Fedor A *et al.* **Breeding novel solutions in the brain: a model of Darwinian neurodynamics** [version 1; referees: awaiting peer review] *F1000Research* 2016, 5:2416 (doi: [10.12688/f1000research.9630.1](https://doi.org/10.12688/f1000research.9630.1))

**Copyright:** © 2016 Szilágyi A *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The author(s) is/are employees of the US Government and therefore domestic copyright protection in USA does not apply to this work. The work may be protected under the copyright laws of other jurisdictions when used in those jurisdictions.

**Grant information:** The research leading to these results has received funding from the European Union Seventh Framework Program (FP7/2007-2013) under grant agreement numbers 308943 (INSIGHT project) and 294332 (EvoEvo project).

*The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.*

**Competing interests:** No competing interests were disclosed.

**First published:** 28 Sep 2016, 5:2416 (doi: [10.12688/f1000research.9630.1](https://doi.org/10.12688/f1000research.9630.1))

## Introduction

The idea that functional selection on a large set of neurons and their connections takes place in the brain during development<sup>1–3</sup> is now experimentally validated<sup>4–7</sup>. As originally portrayed, this process is only one round of variation generation and selection, even if it requires several years. Evolution by natural selection works differently: variants are generated and then selected in iterative rounds. The field of “Neural Darwinism”<sup>1–3</sup> fails to include generation of variants and thus could justifiably be regarded as a misnomer because the process that it describes is not evolutionary in the strict sense<sup>8</sup>. Evidence indicated that the development of the brain is more “constructivist”<sup>9</sup> than pictured by the original selectionist accounts: for example, repeated rounds of neuron addition and loss happen during development<sup>10</sup>. Structural plasticity (synaptic remodelling) is now known to be a lifelong process with implications for memory and learning (e.g. 11,12). The addition and deletion of synapses and neurons takes several hours or days<sup>13</sup>. Our main goal here is to present a proof of principle that bona fide evolutionary dynamics could happen in the brain on a much faster time scale.

Maynard Smith<sup>14</sup> identified multiplication, inheritance and variability as necessary features of evolution. In genetic evolution the variability operators are mutation and recombination. If there are hereditary traits that affect the survival and/or the fecundity of the units, then in a population of these units, evolution by natural selection can take place. While this characterization qualitatively outlines the algorithmic aspect of evolution<sup>15</sup>, concrete realizations require also quantitative conditions: population size cannot be too small (if it is too small, neutral drift dominates over selection<sup>16</sup>) and replication accuracy cannot be too low (if it is too low, hereditary information is lost<sup>17</sup>). Note, that this description says nothing about the nature of the units: they could be genes, organisms, linguistic constructions or anything else.

The proper implementation of an evolutionary process within the nervous system could have major implications for neuroscience and cognition<sup>8,18–25</sup>. A main benefit of neuro-evolutionary dynamics would be that it could harness the parallelism inherent in the nervous system and the redistribution of resources at the same time. The latter process means that hopeless variants are thrown away and are replaced in the “breeding space” by more promising ones<sup>8</sup>. Another important aspect of the process is that it is generative: it could explain where new hypotheses and new policies come from in Bayesian approaches to cognition<sup>26,27</sup> and reinforcement learning<sup>28–30</sup>, respectively. Bayesian inference and natural selection are analogous<sup>31,32</sup> in that candidate hypotheses in the brain (the prior distribution) represent a population of evolutionary units, which are evaluated, or selected, based on the evidence. There is a mathematical isomorphism between the discrete-time replicator equation and Bayesian update<sup>31</sup>. The likelihood function is analogous to the fitness function and the posterior distribution to the selected population. Relations like this suggest that Bayesian update could be one of the pillars of “universal Darwinism”<sup>33</sup>. We believe that convincing models for neuro-evolution could empower Bayesian approaches by providing a mechanism to generate candidate hypotheses.

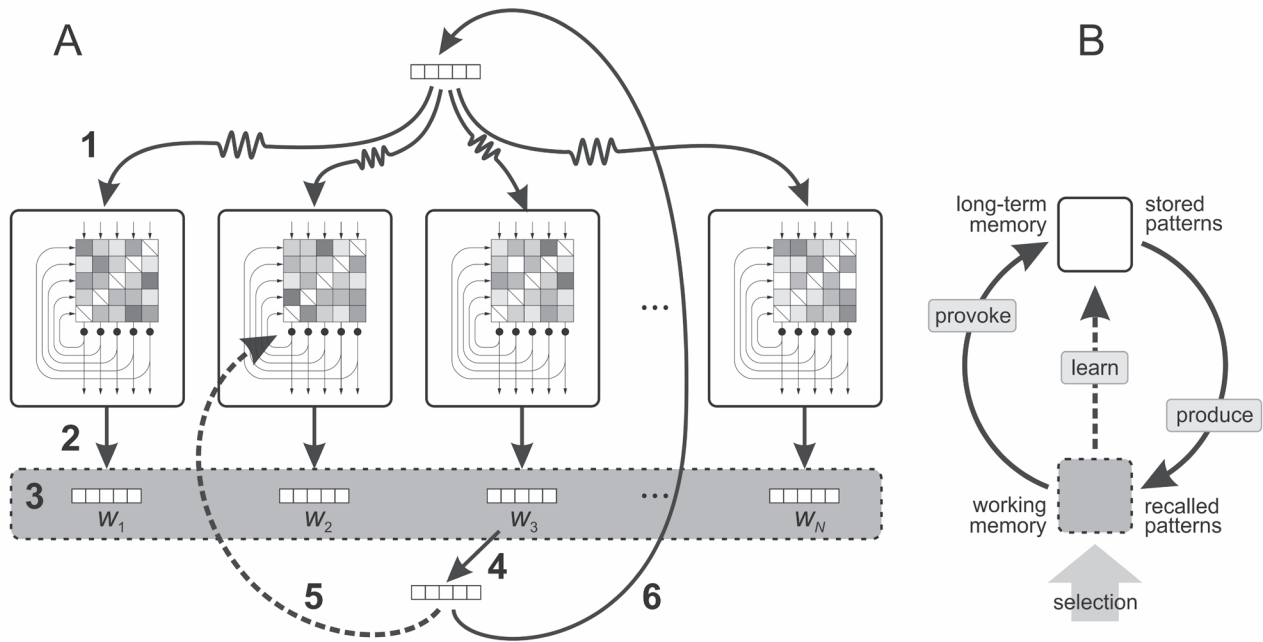
Attractor networks have been used (among others) as models of long-term memory, which are able to complete partial input<sup>34,35</sup>. These networks consist of one layer of units that recurrently connect back to the same layer. The recurrent connections can learn (store) a set of patterns with a Hebbian learning rule. Later, if these patterns or their noisy versions are used to provoke the network, it settles on the original patterns after several rounds of activation updates on the recurrent weights (recall), thus stored patterns act as attractors. It is of high importance that the existence of such networks has been experimentally validated in the visual cortex of awake mice by optogenetic methods<sup>36</sup>. Some versions of the learning rule allow for iterative learning without catastrophic forgetting and enable palimpsest memory. A network with palimpsest memory is able to learn new patterns one-by-one, while sequentially forgetting earlier patterns.

In this paper we describe a model that implements evolution of activation patterns in the brain with the help of attractor networks. We see it as a model of problem solving, which is able to generate new candidate solutions to a problem based on past experiences. Any cognitive problem of the brain is encoded by the activity pattern of neurons. We represent neurons as binary units, being able to continuously maintain firing in one state or the other. A group of neurons at any time therefore has a binary activation pattern. In our model, the units of evolution are these activation patterns, represented as bitstrings. Attractor neural networks can store activation patterns stably for a considerable time in form of corresponding attractors and are able to recall them given the appropriate trigger (Figure 1A). This memory allows for heredity, which is indispensable for Darwinian dynamics (in genetic populations memory is the genotype pool). Attractor neural networks can generate new pattern variants in different ways (corresponding to mutation in a genetic system), see below under Discussion. Owing to memory and pattern generation, the possibility of iterated selection over a population of activation patterns becomes feasible. Our approach thus offers a more natural way to incorporate hereditary dynamics in models of cognitive problem solving at a faster scale that could be provided by, say, structural plasticity (cf. 37). This fast-scale dynamics is missing from Edelmanian Neural Darwinism.

The patterns represent candidate hypotheses or candidate solutions to a problem, which are evaluated based on a fitness function that measures their goodness as a solution. The best patterns are selected and copied (with variation) back to the networks, which in turn generate the next generation of patterns (Figure 1B). Stored patterns constitute the long-term memory; output patterns constitute the working memory (Figure 1B). While pattern generation is a simple recall task, which is only able to reproduce previously learnt patterns, the whole system is able to generate new variants due to noisy recall, spurious patterns (see later), noisy copying of patterns, and iterative learning, thus enabling the evolution of novel solutions.

## Methods

**Recurrent attractor networks.** The basic units in our model are attractor networks. Attractor networks are recurrent neural networks consisting of one layer of units that are potentially



**Figure 1. A) Architecture of multiple attractor networks performing Darwinian search.** Boxed units are attractor networks. Each network consists of  $N$  neurons ( $N = 5$  in the figure, represented as black dots). Each neuron receives input from the top (1) and generates output at the bottom (2). Each neuron projects recurrent collaterals to all other neurons (but not to itself), forming thus  $N \times (N - 1)$  synapses. The weight matrix of the synapses is represented here as a checkerboard-like matrix, where different shades indicate different weights on the connections. Selection and replication at the population level is as follows: 1) Each network receives a different noisy copy of the input pattern. 2) According to its internal attractor dynamics, each network returns an output pattern. 3) All output patterns are pooled in the implicit working memory (grey box with dashed outline), where they are evaluated and a fitness  $w_i$  is assigned to the  $i^{th}$  pattern. 4) The best pattern(s) is selected based on fitness. 5) One of the networks is randomly chosen to learn the pattern that was selected, with additional noise (dashed arrow). 6) The selected pattern is copied back to the networks as input to provoke them to generate the next generation of output patterns. **B) Lifecycle of candidate solution patterns during a cognitive task.** Patterns are stored in the long-term memory as attractors of autoassociative neural networks. When provoked, networks produce output patterns, which are stored in implicit working memory. These patterns are evaluated and selected. Patterns that are good fit to the given cognitive problem can increase their chance to appear in future generations in two possible, non-exclusive ways: 1) selected patterns are retrained to some networks (learning) and 2) selected patterns are used as inputs for the networks (provoking). The double dynamics of learning and provoking ensures that superior solutions will dominate the system. Erroneous copying of patterns back to the networks for provoking and learning and noisy recall are the sources of variation (like mutations).

fully connected. An attractor neural network produces the same (or highly correlated) output whenever the same input is provided (omitting retraining). The pattern that was learned becomes the attractor point of a new basin of attraction, *i.e.* it is the prototype pattern that the attractor network should return. Consequently, an attractor with a non-zero sized basin should also return the same output to different input patterns. However, the amount and type of correlation of input patterns that retrieve the same prototype, *i.e.*, the actual structure of the basin of attraction, is hard to assess, let alone visualize. Still, it is safe to assume that most input patterns correlated with the prototype, produce the same output – the prototype itself.

The Hopfield network is a recurrent artificial neural network with binary neurons at nodes and weighted connectivity between nodes, excluding self-connections. According to the usual convention, the

two states of binary neurons are +1 and -1. In our model, a neuron fires (state +1) if the total sum of incoming collaterals is greater than 0. Accordingly, the update rule has the following form:

$$x_i(t+1) = \text{sgn} \left( \sum_{j=1(\neq i)}^N w_{ij} x_j(t) \right).$$

The original Hebbian (covariance) learning rule has the following form (where  $m$  is the index of the patterns):

$$w_{ij}^0 = 0, \forall i, j \in \{1, 2, \dots, N\},$$

$$w_{ij}^m = w_{ij}^{m-1} + \frac{1}{N} \xi_i^m \xi_j^m.$$

The Hebb rule is both *local* and *incremental*. A rule is local if the update of a connection depends only on the information available on either side of the connection (including information coming from other neurons via weighted connections). A rule is incremental if the system does not need information from the previously learnt patterns when learning a new one, thus the update process uses the present values of the weights and the new pattern. The above update rule performs immediate update of the connection weights (“one shot” process; not a limit process requiring multiple update rounds). The covariance rule has a capacity of  $0.14 N^{58}$ . However, if during learning the system reaches its capacity and further patterns are presented, *catastrophic forgetting* ensues and the network will be unable to retrieve any of the previously stored patterns, forgetting all it has learnt.

To overcome this problem and to preserve the favorable properties of the covariance rule (one-shot, local and incremental updating) Storkey has introduced a palimpsest learning scheme<sup>41</sup> as follows:

$$w_{ij}^m = w_{ij}^{m-1} + \frac{1}{N} \xi_i^m \xi_j^m - \frac{1}{N} \xi_i^m h_j^m - \frac{1}{N} h_i^m \xi_j^m \text{ if } i \neq j,$$

$$w_{ij}^m = 0 \text{ if } i = j,$$

and

$$h_i^m = \sum_{k=1}^N w_{ik}^{m-1} \xi_k^m.$$

Using the above rule, the memory becomes palimpsest (*i.e.* new patterns successively replace earlier ones during learning) with a capacity of  $C = 0.25 N$  (for details and proper definition of palimpsest capacity, see 41).

An interesting feature of some autoassociative neural networks is the appearance of spurious patterns. In some cases, the network converges to a pattern different from any other patterns learnt previously. These spurious patterns can be the linear combination of an odd number of stored patterns:

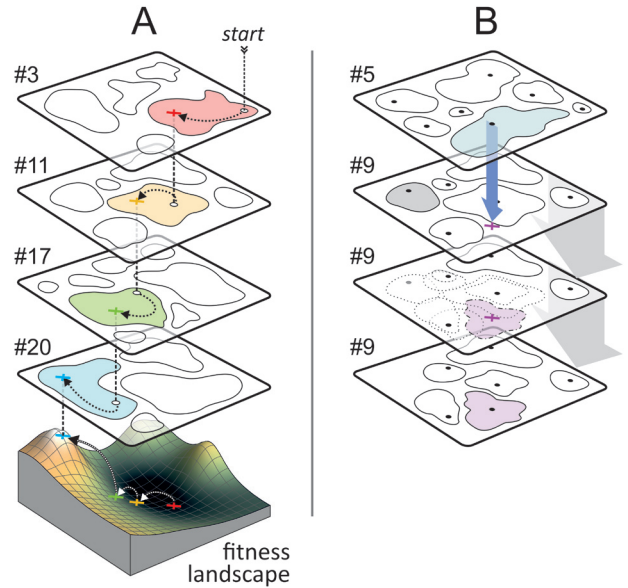
$$\xi_i^{\text{spur}} = \pm \text{sgn}(\pm \xi_i^{m_1} \pm \xi_i^{m_2} \dots \pm \xi_i^{m_S}),$$

where  $S$  is the number of the stored patterns<sup>58</sup>. This effect can be thought of as an effective implementation of a neuronal recombination operator.

**Selection.** For the selection experiment, we used  $N_A$  structurally identical attractor networks, each consisting of  $N$  neurons, implementing Storkey’s palimpsest learning rule.  $N_A = 20$  networks ( $N = 200$ ) were initially trained with random patterns plus a special pattern for each. The 20 special training patterns were as follows. The worst special pattern was the uniform -1, the best special pattern was the uniform +1. Intermediate special patterns had increasing number of +1-s from the left. Fitness was measured as the relative Hamming similarity from the globally best target  $O_{\text{target}}$  (*i.e.* the proportion of +1-s in the pattern). The worst special pattern was trained only to network #1, the second worst to #2, *etc.*, while the best special pattern (which was the target pattern) was trained to network #20. In this scenario, no further training occurred (*i.e.*, the dashed arrows on Figure 1 are not there).

Assuming that the attractor basins of these patterns overlap among networks (Figure 2A) the output of one network will be the cue to trigger one or more close special patterns in other networks. The special patterns ensure that there exists a search trajectory leading from the worst to the best pattern. Starting from any arbitrary initial pattern, if any of the special patterns gets triggered at any time, the system can quickly converge to the optimum.

After initial training, each network received the same random input and generated an output according to its internal attractor dynamics. The output population was evaluated and the best output  $O_{\text{best}}$  was selected based on its fitness. Noisy copies (with  $\mu_p$ , where



**Figure 2. Schematics of attractor networks searching for the global optimum. A) Four time steps of selection, from top to bottom.** At each step, we only show the network that produces the best output (numbered); the rest of the networks are not depicted. In each time step the networks are provoked by a new pattern that was selected from the previous generation of patterns. Different attractor networks partition the pattern-space differently: blobs inside networks represent basins of attraction. At start, the topmost network (#3) is provoked with an input pattern. It then returns the center of the attractor basin which is triggered by the input. When the output of this network is forwarded as input to the next network (#11), there is a chance that the new attractor basin has a center that is closer to the global optimum. If there is a continuity of overlapping attractor basins through the networks from the initial pattern (top) to the global optimum (bottom), then the system can find the global optimum even without learning. **B) Learning in attractor networks.** Network #5, when provoked, returns an output pattern that is used to train network #9 (blue arrow). As the network learns the new pattern, the palimpsest memory discards an earlier attractor (with the gray basin), a new basin (purple shape) forms around the new prototype (purple x) and possibly many other basins are modified (basins with dotted outlines). Black dots indicate attractor prototypes (*i.e.* learnt patterns). With learning, successful patterns could spread in the population of networks. Furthermore, if learning is noisy and a network might learn a slightly different version of the pattern, new variation is introduced to the system above the standing variation. This allows finding the global optimum even if it was not pre-trained to any network. The gray arrow in the background indicates the timeline of network #9.

$\mu$  is the per-bit mutation probability) of  $O_{best}$  were redistributed for each network as new input for the next generation. These steps were iterated until fitness reached the theoretical optimum (*i.e.* the system found special pattern #20). The crucial assumption for selection to work is continuity, namely the possibility that the output of one attractor of one network could fall in a different attractor basin of another network returning an output that is closer to the global optimum than the input was (see Figure 1 and Figure 2).

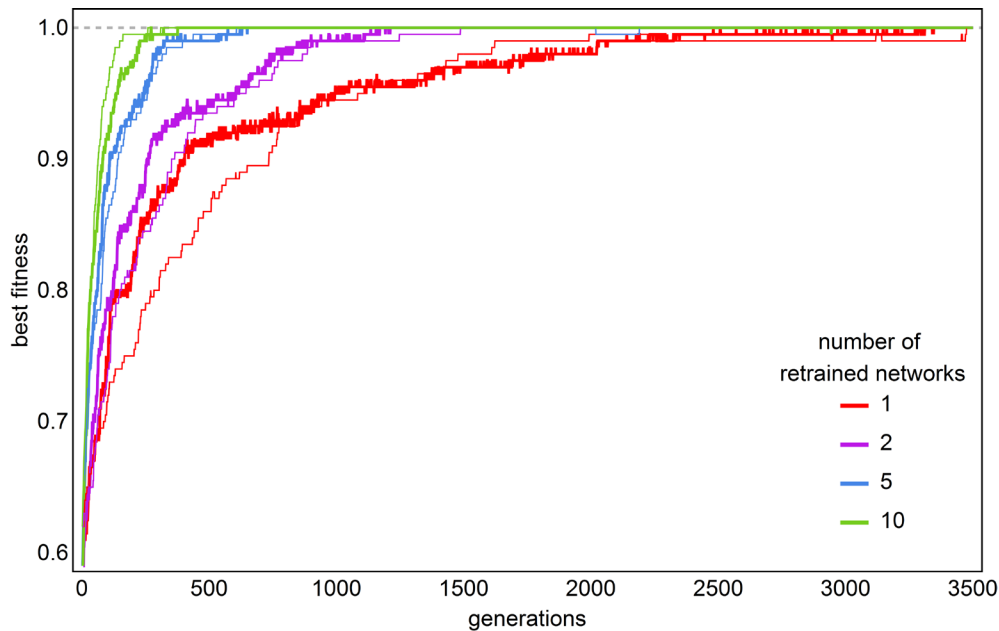
**Evolutionary optimization on a single-peak landscape.** In contrast to purely selective dynamics, in the evolutionary experiment, networks could learn new patterns during the search process. At start, each network was trained with a different set of random patterns. The fitness of a pattern is defined as the relative (per bit) Hamming similarity between the given pattern and an arbitrarily set globally best target pattern  $O_{target}$ . The selection process for the actual best output  $O_{best}$  and redistribution of its noisy copies (with  $\mu_i = 0.005$ ) for input was the same as before. Most importantly, the mutated versions (with  $\mu_T = 0.01$ ) of  $O_{best}$  were also used for retraining  $N_T$  different networks in each generation (see Figure 1): this forms the basis for the Darwinian evolutionary search over attractor networks, as it allows for replication with variation of (learnt) patterns over networks (thin lines in Figure 3).

We have compared the search behavior of our system of attractor networks with a simpler model. In this model networks were represented as abstract storage units, which could store exactly  $C_{fix}$  patterns ( $C_{fix}$  was set to be close to the actual capacity

of networks). When such a storage unit receives an input pattern it simply returns the closest (in Hamming distance) of its stored patterns as output, with additional noise ( $\mu_o = 0.001$ ). The units simulate the almost perfect recall property of attractor networks and effectively approximate attractor behavior. We compared evolution in this simple model with evolution in the system of attractor networks (thick and thin lines in Figure 3).

**Optimization in a changing environment.** In order to test the effect of memory on successive search, we have implemented a periodically changing selective environment, *i.e.*, we periodically changed the fitness function. The environment alternated between  $E_1$  and  $E_2$ , with a stable period length of  $T_E = 2000$ . Each environmental change reset the global optimum: for this scenario, we assumed a uniform +1 sequence for  $E_1$  and its inverse, uniform -1 for  $E_2$  as global optima, and used the relative Hamming similarity as a fitness measure.

In the first phase of the simulation, networks were allowed to learn in each environment for a total of  $T_{nolearn} = 12000$  generations (three periods per environments). Afterwards, learning was turned off to test the effect of memory. To make sure that the optimal pattern was not simply carried over as an output pattern from the previous environment but was recalled from memory, the input patterns were set to random patterns (instead of inheriting the previous output population) at the start of each new environmental period after  $T_{nolearn}$ . This ensures that the population could only maintain high fitness afterwards in an environment if the optimum



**Figure 3. The effect of retraining on the speed of evolution.** Lines represent the evolution in four different populations, where a different number of networks were retrained. Each population consisted of 10 networks (see the rest of the parameters under the Methods section). Thin lines: stochastic attractor dynamics; thick lines: simulated attractor dynamics (abstract networks always return the stored attractor prototype that is closest to the actual input, with 0.001 per bit probability noise; capacity to store  $C_{fix} = 30$  patterns,  $\mu_o = 0.002$ ). Parameters:  $N = 200$ ,  $N_A = 20$ ,  $\mu_T = 0.01$ ,  $\mu_i = 0.005$ , elitist selection, keeping the best one only from each output generation; retraining selects random networks (never the same in a given generation). Fitness is the relative Hamming similarity to the global optimum.



was stored *and* could be successfully recalled (see Figure 4). In order to assess the memory of a network, we also measured the distance between the actual best output of the population and the closest one of the set of previously learned patterns within the same network (as different networks have different training history). A small distance indicates that the network outputs a learned pattern from memory (*i.e.* recalls it) instead of a spurious pattern.

For this scenario, we introduced a different selection method (also used in the next section). Each network in the population produces an output according to its internal attractor dynamics and the input it received from the previous generation. From all output sequences one was randomly chosen and mutated ( $\mu_R = 1/N$  per bit mutation rate). If the mutant had a higher fitness than the worst of the output pool, the worst pattern was replaced by it (*elimination of the worst*). Furthermore, in the case of a superior mutant, it was also trained to  $N_T$  number of different networks. Lastly, the resulting output population is shuffled and fed to the networks as input in the next generation (except when the environment changes and input is reset externally).

**Optimization on a difficult landscape.** To investigate the applicability of this optimization process, we adopted a complex, deceptive landscape with scalable correlation, and also modified the selection algorithm introduced above. We used the general building-block fitness (GBBF) function of Watson and Jansen<sup>38</sup>. According to the GBBF function, each sequence of length  $N$  is partitioned into blocks of uniform length  $P$ , so that  $N = PB$

( $P, B \in \mathbb{Z}^+$ ) where  $B$  is the number of blocks. For each block,  $L$  arbitrarily chosen subsequences are designated as local optima, with randomly chosen but higher-than-average subfitness values. The overall fitness  $F(G)$  of a pattern  $G$  (“genotype”) is as follows:

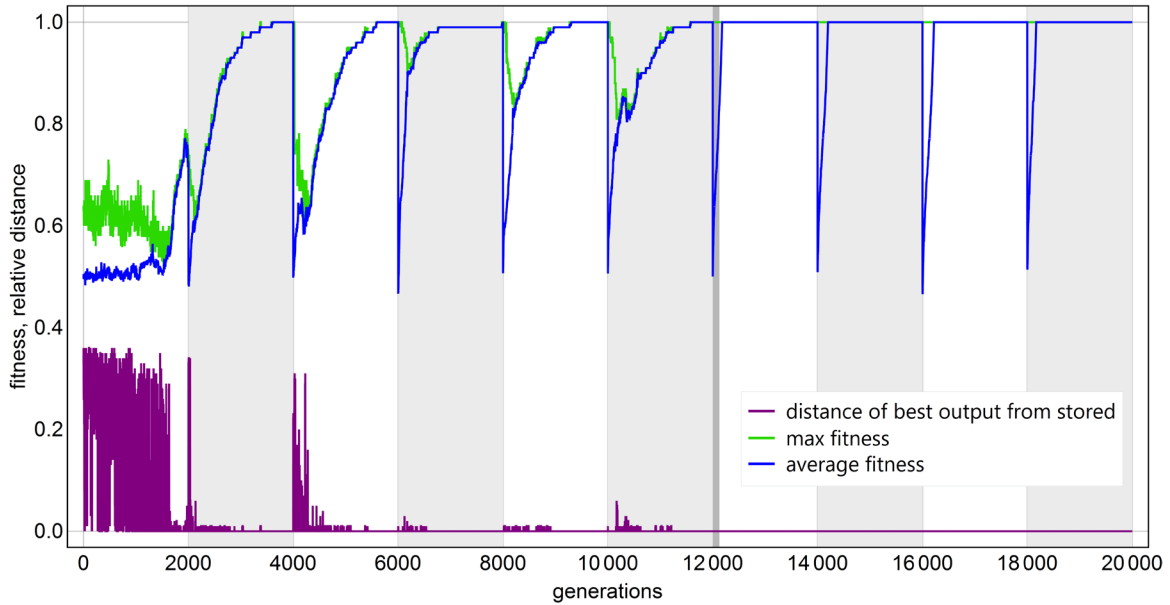
$$F(G) = \sum_{i=1}^B f(g_i),$$

$$f(g_i) = \sum_{j=1}^L c(g_i, t_j),$$

$$c(g, t_j) = \begin{cases} w_j, & \text{if } d(g, t_j) = 0 \\ \left(1 + d(g, t_j)\right)^{-1}, & \text{otherwise} \end{cases},$$

where  $f(g_i)$  is the fitness contribution of the  $i$ th block in the pattern,  $t_j$  is the  $j^{\text{th}}$  local optimum of length  $P$  (all  $L$  different optima are the same for each block in our experiments) with subfitness value  $w_j > 1$ , and  $d$  is the Hamming distance. Consequently, this landscape has many local optima, a single global optimum and a highly structured topology. Furthermore, since there are no nonlocal effects of blocks, each block can be optimized independently, favoring a metapopulation search.

Accordingly, in this experiment, we introduced multiple populations of attractor networks. Each population of  $N_A$  attractor neural networks forms a deme and  $N_D$  demes are arranged in a 2D square



**Figure 4. Fitness and recall accuracy over periodically alternating environments.** Blue: average fitness; green: best fitness; purple: distance of the best output of the population from the closest one stored in memory (for details, see main text). Grey and white backgrounds represent the changing environment: We alternated two global optima at every 2000<sup>th</sup> generation. After the 12000<sup>th</sup> generation, we turned off learning (thick vertical line) and set the input to random patterns after each changing of the environment. Parameters:  $N_A = 100$ ,  $N = 100$ ,  $N_T = 40$ , fitness is the relative Hamming similarity to the actual optimum.

lattice of Moore neighborhood (all of the eight surrounding demes are considered neighbors). Demes might accept output sequences from neighboring demes with a low probability  $p_{migr}$  per selection event; this slow exchange of patterns can provide the necessary extra variability for recombination. These demes correspond to the groups of columns in the brain.

Networks in the deme are the same as those used in previous experiments. However, selection is modeled in a different way, similar to the selective dynamics outlined in 38. In turn, we only give a brief description. Given a deme, each network produces an output according to its internal attractor dynamics and the input it received from the previous generation. Output sequences are pooled and either one or two is randomly chosen for mutation or recombination, respectively (*i.e.* *no elitist selection*). With probability  $p_{rec}$ , two-point recombination is performed of the two selected partners, with  $1-p_{rec}$  probability, a single selected sequence is mutated, with  $\mu_R = 1/N$  per bit mutation rate. With  $p_{migr}$  probability, the recombinant partner is chosen from another neighboring deme instead of the focal one. Next, the output(s) of recombination or mutation are calculated: if the resulting sequence (any of the two recombinants or the mutant) has a higher fitness than the worst of the output pool, it is replaced by the better one (*elimination of the worst*). Furthermore, in the case of a superior mutant or recombinant, it is also trained to  $N_T$  number of different networks within the deme. Lastly, the resulting output population is shuffled and fed to the networks as input in the next generation. Each deme is updated in turn according to the outlined method; a full update of all networks in all demes constitutes a generation (*i.e.* a single time step).

The GBBF landscape was set up identically to the test case in 38, as follows. For each block uniformly, two target sequences of length  $P$ ,  $T_1$  and  $T_2$ , were appointed.  $T_1$  is the uniform plus-one sequence  $T_1 = \{+1\}^P$  and  $T_2$  is alternating between -1 and +1 ( $T_2 = \{-1, +1\}^{P/2}$ ). According to the fitness rule (Equation 5–Equation 6 in 38 and Equation 1–Equation 3 above), the best subfitness of each block in a sequence can be calculated and the sum of all the subfitness values is the fitness of the global optimum sequence. Thus for sake of simplicity, we used relative fitness values with the global optimum (the uniform +1 sequence) having maximal fitness 1. The sequence(s) with lowest fitness always have a nonzero value.

The source code of all models and data presented in this paper is freely available as a supplement to this paper.

## Results

**Selection.** We should distinguish between two processes: (i) search without learning among the stored patterns to find the best available solution (*i.e.*, selection without step 5 on Figure 1A), and (ii) search with learning: retrain one or more networks with the selected and mutated patterns (Figure 1A with step 5). The first is a purely selectionist approach because it cannot generate heritable variants, while the second implements Darwinian evolution because learning changes the output behavior of the networks, thus they generate new patterns. First, we analyze the strictly selectionist version, and then the evolutionary version of the model.

In the selectionist version we pre-trained each network with a random set of patterns (excluding the target pattern) and started by provoking them with a different random input. Each network produced an output pattern according to its own attractors and then the best pattern was selected. This pattern was used in turn to provoke the networks in the next generation, and so on. This search has found among all the available stored (pre-trained) patterns the one with the highest fitness; it could not find the global optimum, as the networks were not pre-trained with it and there was no way for new variants to appear in this simulation.

Next, we specifically composed the sets of pre-training patterns: each network was pre-trained with random patterns as before but also with one special pattern. This set of special patterns (in which individual patterns can be ordered according to gradually increasing fitness values) delineate a route to the optimum through overlapping basins of attractors in different networks (see Figure 2A) so that we can test whether in this simplified case the algorithm converges quickly to the optimum. The first population was initiated with the special pattern that was farthest from the optimum. We have found that the selected output gets closer to the optimum in each generation, but the optimization process is saltatory: it skips over many intermediate neighboring special patterns (and thus networks). This is due to the fact that attractor basins of neighboring special patterns were highly overlapping. For example, in Figure 2A, the stored special pattern of network #3 is in the basins of stored special patterns of networks #4–#11, and since the stored pattern of network #11 is closest to the optimum, networks #4–#10 were skipped. A typical sequence of networks generating the actual best output is: #3, #11, #17 and #20 (of 20 networks; for actual parameters, see Figure 2A).

**Evolution.** Learning new patterns as attractors (Figure 2B) allows networks to adapt to the problem and perform evolutionary search. The results of the evolutionary experiments clearly prove that a population of attractor networks can implement evolutionary search in problem spaces of different complexity (*i.e.* different levels of correlation and deceptiveness).

**Evolution on a simple fitness landscape.** In this scenario, neither the global optimum nor a route toward it is assumed to pre-exist in the system as in the selectionist experiments: networks are pre-trained only with random patterns. Even under these stringent conditions, we have found that the system can converge to the global optimum, and this convergence is robust against a wide range of mutation rates. Our simplified abstract model, which always returns the stored prototype that is closest to an input, behaves qualitatively the same way (see Figure 3). The speed of convergence to the optimum is mainly affected by the number of retrained networks (Figure 3): as we increase the number of networks that are retrained we find a faster fitness increase, albeit with diminishing returns. Mutation has an optimal range in terms of the speed of evolution. On one hand, if mutation rate is too low evolution slows down, because there is not enough variation among patterns. On the other hand, if mutation rate is too high it hinders evolution as the offspring is too dissimilar to the parent and cannot exploit the attractor property of the system. When mutation rate is zero, the source of variation is only the probabilistic input-output behavior of the networks due to



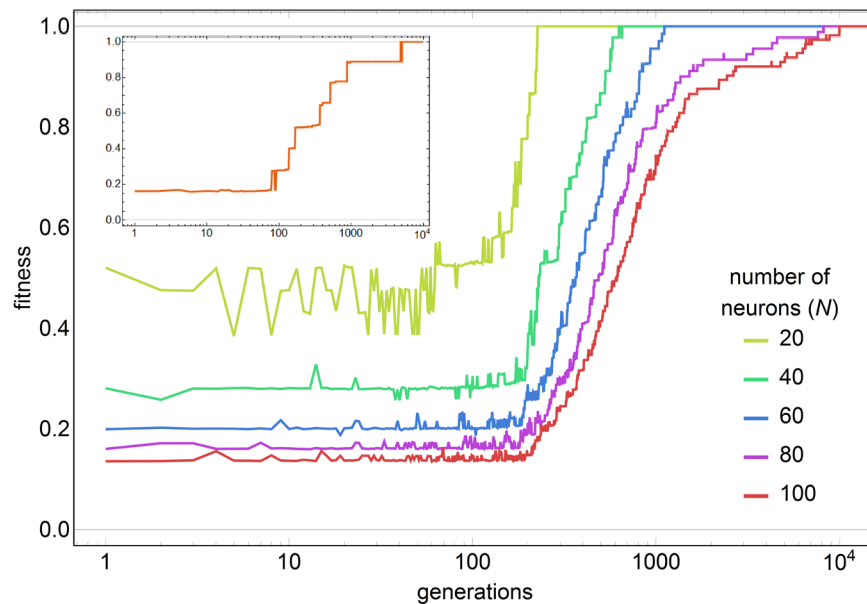
their asynchronous update and the appearance of spurious patterns when the input is too far from the stored patterns.

While the attractor networks have memory, due to the monotonic, single-peak nature of the fitness landscape there is no need to use it: the system works almost equally well if the networks only store the last trained pattern (*i.e.*, weights are deleted before each learning event). Next, we present experiments where both the attractor property and the palimpsest memory of the networks are used.

**Evolution in a changing environment.** In this experiment we alternated two environments: in every 2000th generation the target pattern (the optimum), against which fitness was measured, was changed. From an evolutionary point of view, this can be perceived as a changing environment, whereas from a cognitive point of view, this procedure simulates changing task demands. Figure 4 shows that the system found and learnt the optima of each of the two environments separately. Then, after generation 12000, we switched off learning. The fact that networks are nevertheless able to recall the target pattern right after the environmental change proves that they use previously stored memories. After we switched off learning, we used random patterns to provoke networks at the first generation of each new environment. A single network that can recall the optimum from the random input is enough to produce a correct output that is amplified by selection for the next generational input, ultimately saturating the population with optimal patterns. This experiment effectively proves that a system of attractor networks can reliably recall earlier stored solution patterns, therefore solves the problem faster in an alternating environment than a system without long-term memory.

**Evolution on a difficult fitness landscape.** The previous evolutionary experiment (where search was on a single-peak fitness landscape with a single population of networks) is a proof of principle of the effectiveness of our evolutionary algorithm. In order to assess the capacity of population search of attractor networks, we introduce a considerably harder fitness landscape with higher dimensionality, where the deceptiveness of the problem can be tuned. The GBBF fitness landscape of 38 provides a method to easily generate scalable and complex landscapes with many deceptive local optima. The complexity of the problem requires the introduction of multiple interacting populations of networks. Though explicit spatial arrangement of the networks is not required to solve the problem, we have nevertheless included it in our implementation to imitate real spatial arrangement of neurons in the brain. Locality allows the exchange of information among neighboring populations (*i.e.* recombination) that is essential to solve the GBBF problem (or similar deceptive problems) in a reasonable time.

We have investigated the performance of search in a metapopulation with different problem sizes (pattern lengths; see Figure 5). Results indicate that despite the vastness of the search space, the metapopulation is always able to converge to the global optimum, given enough time. The most complex landscape of 100-bit patterns is of size  $2^{100}$  with one global optimum and a huge number of local optima. The metapopulation consists of  $10^5$  neurons (100 populations of 10 networks each with 100 neurons per network) and can find the single global optimum in  $\sim 10^4$  time steps. The limit of further increasing the problem size is in the computational capacity of our resources.



**Figure 5. Convergence of the actual best fitness of the metapopulation with increasing problem size  $N$  on general building block function landscape.** Each curve is an average of 10 independent iterations.  $N_D = 10 \times 10$  demes,  $N_A = 10$  networks per deme,  $N$  neurons per network, patterns of length  $N$  are partitioned to blocks of size  $P = 10$  ( $B = N/P$  blocks per pattern),  $p_{rec} = 0.1$ ,  $\mu_R = 1/N$ ,  $p_{migr} = 0.004$ ,  $N_T = 5$ . Note the logarithmic x axis. Inset: single simulation at  $N = 80$ ,  $B = 8$  (other parameters are the same). Plateaus roughly correspond to more and more blocks being optimized by finding the best subsequence on the building-block fitness landscape.

## Discussion

**Summary of results.** Attractor networks can be used to implement selection, replication and evolutionary search. As a proof of principle, we showed that attractor networks find the global optimum in a purely selectionist model (*i.e.* without learning) if they are pre-trained with attractors that overlap in their basins and lead to the optimum. The population can effectively select over the standing variation of all stored patterns and find the trajectory to the (single) peak of the fitness landscape (see [Figure 2B](#)). Furthermore, if learning is allowed during search, the relative frequency of good patterns (those closer to the optimum) can be increased by re-training networks with such patterns, so that they are stored in the long-term memory in more copies ([Figure 3](#)). Overwriting an older memory trace with a new pattern corresponds to a copying operation, potentially with mutations. A particularly interesting aspect of a population of attractor networks in the given coupling is that even if learning is not erroneous, the Lamarckian nature of inheritance of patterns (as output  $\rightarrow$  memory  $\rightarrow$  output; see [Figure 1B](#)) means that there is room for heritable variation to emerge at other stages of the cycle, thus implementing Darwinian dynamics.

The explicit benefit of memory manifests in a periodically changing environment. In a single, stable environment, memory is not very useful because the attractor property acts against exploring variation, and networks might be even slower than gradient hill-climbers (*i.e.* searchers who generate mutant output blindly and only take a step on the landscape if the output is better than the input). However, in a periodically changing environment, attractor networks with memory are able to recall the actual global optimum if they have already encountered the given environment in the past and stored its optimum; hill-climbers or naïve attractor networks lacking the necessary memory would have to perform the search all over again. Attractor network can recall the appropriate pattern even if there is no initial cue for the population to know which environment it is in. A network with memory can complete a partial cue and thus can recall the global optimum. After learning has ceased, it is enough to only have a few networks in the population that can recall the optimum from random cues that are accidentally close to the actual optimum (maximal fitness is immediately 1 in the population, see green curve in [Figure 4](#)). Selection then simply amplifies these optimal output patterns in the population (as output is fed back as input in the next generation) until all networks receive optimal patterns as input. At that point, average fitness also reaches the maximum ([Figure 4](#), blue curve). A network without memory would have to search for the new optima in each environment over and over again, finding the whole uphill path on the landscape.

We also proved that a metapopulation of attractor networks can successfully optimize on a complex, correlated landscape of higher dimensionality. This is a notoriously hard optimization problem (*cf.* [38](#)) as hill-climbers can easily get stuck in deceptive local optima. The spatial organization of attractor networks resembles the spatial organization of neural structures of the cortex and it allows parallel optimization of subproblems. By this independent optimization of solution parts, local populations can exchange successful partial solutions with each other and form recombinants that are necessary to solve such complex problems in reasonable time.

**Evolutionary context.** We have chosen attractor networks to demonstrate Darwinian neurodynamics because (i) the search for existing solutions uses the same architecture for generating, testing and storing novel ones; (ii) stored patterns help evolutionary search by readily employing related past (learned) experience, and (iii) the particular choice of Storkey's model naturally results in some new recombinant patterns. This is an important point because, as we know from population genetics<sup>39</sup>, recombination speeds up the response to selection by creating new variants<sup>40</sup>.

Our choice of implementation of attractor networks, following Storkey's model<sup>41</sup>, is based on three important aspects: (i) it has palimpsest memory, so that it can learn new and forget old patterns without catastrophic forgetting, as happens in Hopfield and other networks; (ii) its attractors are roughly of equal size and are well-structured according to a Hamming distance measure, and (iii) unlike most other neural networks, it is able to store correlated patterns. The downside is that these networks require full connectivity, which is neurally unrealistic. However, its functional properties reflect well what we know of long-term memory in the brain, which is enough for a proof of principle of an evolutionary implementation of neuronal function. To our knowledge no model exists in the literature that could satisfy all requirements above and that, at the same time, works with diluted connectivity<sup>42</sup>.

It is important to clarify that the units of evolution in our proposed Darwinian neurodynamics system are the activation patterns: they are copied potentially with errors, selected and amplified. However, patterns live in two stages: in the working memory and in the long-term memory (*cf.* [Figure 1](#)). This implies different inheritance methods (routes to pass on information) from what is expected in a purely genetic system. Changed attractor prototypes imply changed outputs, just like a mutated genotype implies a different phenotype. However, in our proposed system, changes made to output patterns (by mutation) can also be "inherited" by the stored attractor prototypes via learning. Furthermore, there is another difference in the dynamics, explained in turn.

Darwinian evolution is often described as a parallel, distributed search on a fitness landscape<sup>8</sup>. The population, as an amoeba, climbs the peaks of the landscape via iterative replication and selection in successive generations. The attractors, however, impose a different mode of evolution, because they simply return the prototype pattern closest to the input, even if it is less fit than the input pattern itself. Consequently, attractor networks work against variability and slow down hill-climbing. However, attractor networks resist fitness increase only half of the time on average; the other half of the time they effectively push inferior patterns uphill in the fitness landscape at a speed much higher than that expected for the same (reduced) amount of genetic variation. Consequently, attractor networks can facilitate evolution<sup>21</sup>.

We stress the importance of evolutionary combinatorial search. In cases where *ab initio* calculation of molecular functionality is impossible, artificial selection on genetic variants has proven to be an extremely useful method to generate molecular solution to functional problems, as experiments on the generation of catalytic RNA molecules (ribozymes) illustrate (see [43](#) for a recent review). By the same token when a brute force numerical calculation of a

combinatorial functionality problem is impossible for the brain, given the adequate architecture it could (and we suggest it does) use an evolutionary search mechanism, as shown in this paper.

**Implementation in the brain.** It is of primary importance that all the components in our ‘algorithmic diagram’ (Figure 1) can be implemented by mechanisms known in the brain. It is likely that the cortical hypercolumn<sup>44</sup> behaves like an (at least one) attractor network. The reciprocal connections between the long-term and working memory networks are assumed to be like those in Figure 3 in <sup>45</sup>. We propose that, first, the reinforcing signal from the basal ganglia via the thalamus keeps active the good candidate pattern solutions in the rewarded auto-associative network and, second, that the latter sends a copy of the active pattern to (unconscious) working memory for eventual redistribution. When there is an increase in the quality of a solution (fitness increase) or when a local or global optimum is reached, the central executive elicits the transmission of a copy of the solution to the conscious memory<sup>46</sup>.

Our proposed mechanism relies on information transmission of patterns between cortical groups and relevant subcortical structures with variation, and in this way it differs from all previous models. A discussion of timescales is in order. Without learning newly generated variants, the selective mode would require about the same time as suggested by <sup>47</sup> as the “cognitive cycle” (based on data), but without perception at the beginning of the cycle, *i.e.* it would be in the 160–310 ms range. In the evolutionary mode, learning of new variants is required which would take more time. A conservative estimate for the reliable expression of changed synaptic weights is between seconds to minutes<sup>48,49</sup>. The second scale would allow several dozen cycles/generations per minute—a very good number in artificial selection experiments. A more accurate estimate of timing will require a fully neuronal model of our proposed mechanism.

Copying of information is necessary for evolutionary processes: this is where previous approaches<sup>22–24,50</sup> have been too artificial. There are four well known instances where scholars invoke copying of information in the brain: (i) the transfer of information from short to long-term memory<sup>35,51,52</sup>, (ii) the transfer of information from specialized cortical areas to working memory<sup>53</sup> or to the global workspace<sup>54,55</sup> and, finally (iii) the possible copying of patterns from working memory to conscious processing<sup>56</sup>. Undoubtedly, all these approaches require accurate information transfer between different brain components<sup>57</sup>.

There are three sources of variation in our system: (i) due to the finite size of our networks and asynchronous update of the neurons, the output patterns show some variation even if a network is repeatedly provoked by the same input pattern, (ii) acknowledging the noisiness of neuronal transmission we introduce variation akin to mutations when patterns are transmitted among the blocks of the model, and finally (iii) we have realized that “spurious patterns”<sup>58</sup> emerge as by-products of previously trained patterns and they might act as (new) recombinants of learnt patterns that facilitate the evolutionary search.

The non-conscious or implicit working memory, which has received considerable attention lately<sup>46,56</sup>, is crucial for our proposed mechanism. Irrespective of whether working memory overlaps with the conscious domain<sup>59</sup> or not<sup>56</sup> (in the latter case a ‘conscious copy’ must be sent from working memory to conscious access), the important factor is that the bound on the number of patterns that can be held in the *unconscious* part of the working memory is larger than that of the conscious working memory<sup>59</sup>. In other words, our mechanism suggests that the total storage capacity of the unconscious network population is much higher than that of the conscious one. Crucially, there is support for this requirement: there is evidence that the central executive function of working memory is not restricted to the conscious domain either<sup>46</sup>. The relatively large capacity of (unconscious) working memory can hold not one, but several patterns selected by the cortex-striatum-basal ganglia loop. This type of selection can be realized by a winner-share-all (WSA) mechanism<sup>60</sup>.

The latter point requires special attention. The reader is referred to the recent review by <sup>61</sup> on models of action selection and reinforcement learning. We wish to make a few critical points in this regard. First, as we are considering problem solving that unfolds its capacity online, there is no reason to select one pattern, since the interim patterns are not alternative actions but only candidate solutions. They can be turned into actions sometimes during, or only at the very end, of the evolutionary search. Weak lateral inhibition within the evaluation mechanism enhances value differences in selection, but a single winner is not selected<sup>60,61</sup>. Second, parallelism of the evolutionary approach loses considerable power if the evaluations are not done in parallel, and if poor solutions cannot be replaced by better solutions in the storage. (In a subsequent study we shall show that the number of parallel evaluations are allowed to be considerably smaller than population size but also that purely serial evaluation of candidates is a killer). Third, it is perfectly possible that the WSA part is implemented by the cortex rather than the striatum (cf. <sup>61</sup>): we are agnostic on this point for the time being. (Admittedly that option would require a different version of the full model). Fourth, we maintain that parallel survival of a number of candidates should happen, and the mechanism for this might have evolved with selection for complex (offline) problem solving. Fifth, it is well possible that WSA is gradually reduced towards WTA (one winner takes all) during evolutionary search in the brain: this would also guard against premature convergence early and fast convergence towards the end of the search.

To sum up, we have seen that a process analogous to natural selection can be rendered into a neuronal model employing known neurophysiological mechanisms. Now we discuss relations to some other publications and outline future work.

**Related work.** Several examples show that evolution with neurodynamics can be more powerful than either of the components alone. Fernando *et al.*<sup>25</sup> proved that the path evolution algorithm – which includes both elements of structural plasticity<sup>62,63</sup> and iterative generation of variation – is more powerful in several regards than classical genetic algorithms. Fernando *et al.* have also shown that replication combined with Hebbian

learning is more powerful than classical natural selection in a model of mechanistic copying of binary activity<sup>22</sup>. De Vladar and Szathmáry provided proof that the synergy between selection and learning results in increased evolvability; also they pointed out that synaptic plasticity helps escaping impasses and build circuits that are tailored to the problem<sup>21</sup>. Finally, in a recent model Fernando and his colleagues have used autoencoders for the generation of “neuronal genotypes”<sup>64</sup>. Since autoencoders produce compressed representations of the input, we expect them to successfully replace the identity function (*i.e.* bit by bit copying, as in DNA replication). Indeed, applying this neural component within the context of a genetic algorithm turned out to be rewarding.

Unless the envisaged information transfer is accurate enough in space and time, the evolutionary dynamics breaks down. Similar to genetic evolution, where the error threshold<sup>17</sup> had to be raised before long informative genomes could arise by evolving adaptations for more accurate copying<sup>65</sup>, in the neuronal context the element of accuracy was raised by Adams<sup>18</sup>. In his “Hebb and Darwin” paper Adams talks about synaptic replication and synaptic mutation as important ingredients for a Darwinian view of the brain. Synaptic replication means either the strengthening of an existing synapse, or the making of a new synapse between two neurons that already have one synapse between them. Adams’ is an important insight: evolutionary dynamics does not need copying for selection (scoring or strengthening is enough), but it needs copying with errors to test the new variants against the others. Synaptic mutation happens when a neuron grows a synapse towards a neighboring neuron with which previously it had no contact. Interestingly, these thoughts preceded the burst of interest in structural synaptic plasticity (SSP;<sup>62,63</sup>). Following his expansion-renormalization model for SSP<sup>66</sup> Kilgard observes that when SSP is used for learning something new, this could be regarded as a Darwinian mechanism<sup>37</sup>, as it generates and tests variations in successive rounds, based on what is already there (unlike the original models of “neural Darwinism”). Kilgard’s mechanism has not been formalized yet (although see 21), but the path evolution model<sup>25</sup> bears some relationship to it.

We share the view of Eliasmith<sup>53</sup> that the cortex/basal ganglia/thalamus/cortex loop plays a crucial role not only in elementary action selection but also in symbolic reasoning. We conjecture that non-primate animals (in particular mammals and birds) employ the same (or at least an analogous) loop in order to retrieve old and to innovate new solutions, in a similar way as we have shown using our elementary model.

Another view to which we feel strongly related to is that of Bayesian models that advocate “theory learning as stochastic search in a language of thought”<sup>27</sup>. We are reasonably confident that we have found a candidate mechanism for the search process. If true, the rugged learning landscape in Figure 3 of 27 can be directly interpreted as the fitness landscape of our neuro-evolutionary model. A task for the future is to work out the

explicit relations in detail. We note again the formal link between Bayesian inference and evolutionary selection<sup>31,32</sup> mentioned in the Introduction. Our mechanism (Figure 1) could in principle implement, with appropriate modifications, an estimation of distribution algorithm (EDA). The population-based incremental learning (PBIL) algorithm consists of the following steps<sup>67</sup>: (i) Generate a population from the probability vector; (ii) Evaluate and rank the fitness of each member; (iii) Update the probability vector based on the elite individual; (iv) Mutate the probability vector; (v) Repeat steps (i-iv) until a finish criterion is met. EDA can work better than copying algorithms making it an interesting line to pursue.

Future work will be to link our recurrent model with the feedforward autoencoder model of Fernando *et al.*<sup>64</sup>, since the latter can generate interesting genotypes (better substrates for selection) due to the emerging compressed representations of the inputs.

As two experts aptly remark: “The Bayesian brain falls short in explaining how the brain creates new knowledge” (68 p. 9). We suggest that neuronal evolutionary dynamics might serve as a remedy.

### Data and software availability

Zenodo: IstvanZachar/Neurodynamics: Publication release, doi: 10.5281/zenodo.154113<sup>69</sup>.

The algorithm described in the paper is also available on GitHub at <https://github.com/IstvanZachar/Neurodynamics>.

### Author contributions

ESz conceived the model. ASz and IZ coded the model. ASz, IZ, AF and HPdV contributed to the model and designed the experiments. ASz and IZ run the experiments and analyzed the data. All authors contributed to writing the manuscript.

### Competing interests

No competing interests were disclosed.

### Grant information

The research leading to these results has received funding from the European Union Seventh Framework Program (FP7/2007-2013) under grant agreement numbers 308943 (INSIGHT project) and 294332 (EvoEvo project).

*The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.*

### Acknowledgements

We are thankful to Luc Steels, Chrisantha Fernando, Mauro Santos, and Thomas Filk for useful comments and discussions.



## References

1. Changeux JP: **Neuronal man: The biology of mind**. Princeton, NJ: Princeton University Press; Translated by Garey, L. 1985.  
[Reference Source](#)
2. Changeux JP, Courrége P, Danchin A: **A theory of the epigenesis of neuronal networks by selective stabilization of synapses**. *Proc Natl Acad Sci U S A*. 1973; 70(10): 2974–2978.  
[PubMed Abstract](#) | [Free Full Text](#)
3. Edelman GM: **Neural Darwinism. The theory of neuronal group selection**. New York: Basic Books; 1987.  
[Reference Source](#)
4. Williams RW, Rakic P: **Elimination of neurons from the rhesus monkey's lateral geniculate nucleus during development**. *J Comp Neurol*. 1988; 272(3): 424–436.  
[PubMed Abstract](#) | [Publisher Full Text](#)
5. O'Leary DD: **Development of connectational diversity and specificity in the mammalian brain by the pruning of collateral projections**. *Curr Opin Neurobiol*. 1992; 2(1): 70–77.  
[PubMed Abstract](#) | [Publisher Full Text](#)
6. Rabinowicz T, de Courten-Myers GM, Petetot JM, *et al.*: **Human cortex development: estimates of neuronal numbers indicate major loss late during gestation**. *J Neuropathol Exp Neurol*. 1996; 55(3): 320–328.  
[PubMed Abstract](#) | [Publisher Full Text](#)
7. Miller-Fleming TW, Petersen SC, Manning L, *et al.*: **The DEG/ENAC cation channel protein UNC-8 drives activity-dependent synapse removal in remodeling GABAergic neurons**. *eLife*. 2016; 5: pii: e14599.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
8. Fernando C, Szathmáry E, Husband P: **Selectionist and evolutionary approaches to brain function: a critical appraisal**. *Front Comput Neurosci*. 2012; 6: 24.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
9. Quartz SR, Sejnowski TJ: **The neural basis of cognitive development: a constructivist manifesto**. *Behav Brain Sci*. 1997; 20(4): 537–556; discussion 556–96.  
[PubMed Abstract](#) | [Publisher Full Text](#)
10. Bandeira F, Lent R, Herculanu-Houzel S: **Changing numbers of neuronal and non-neuronal cells underlie postnatal brain growth in the rat**. *Proc Natl Acad Sci U S A*. 2009; 106(33): 14108–14113.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
11. Caroni P, Donato F, Müller D: **Structural plasticity upon learning: regulation and functions**. *Nat Rev Neurosci*. 2012; 13(7): 478–490.  
[PubMed Abstract](#) | [Publisher Full Text](#)
12. Bernardinelli Y, Nikonenko I, Müller D: **Structural plasticity: mechanisms and contribution to developmental psychiatric disorders**. *Front Neuroanat*. 2014; 8: 123.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
13. Tetzlaff C, Kolodziejewski K, Markelic I, *et al.*: **Time scales of memory, learning, and plasticity**. *Biol Cybern*. 2012; 106(11–12): 715–726.  
[PubMed Abstract](#) | [Publisher Full Text](#)
14. Maynard Smith J: **The problems of biology**. USA: Oxford University Press; 1986.  
[Reference Source](#)
15. Maynard Smith J: **Genes, memes, and minds**. The New York Review of Books. 1995; 42(19): 46–48.  
[Reference Source](#)
16. Kimura M: **The Neutral Theory of Molecular Evolution**. Cambridge: Cambridge University Press; 1983.  
[Publisher Full Text](#)
17. Eigen M: **Selforganization of matter and the evolution of biological macromolecules**. *Naturwissenschaften*. 1971; 58(10): 465–523.  
[PubMed Abstract](#) | [Publisher Full Text](#)
18. Adams P: **Hebb and Darwin**. *J Theor Biol*. 1998; 195(4): 419–438.  
[Publisher Full Text](#)
19. Calvin WH: **The brain as a Darwin Machine**. *Nature*. 1987; 330(6143): 33–34.  
[PubMed Abstract](#) | [Publisher Full Text](#)
20. Calvin WH: **The cerebral code: thinking a thought in the mosaics of the mind**. Cambridge, MA: MIT Press. 1996.  
[Reference Source](#)
21. de Vladar HP, Szathmáry E: **Neuronal boost to evolutionary dynamics**. *Interface Focus*. 2015; 5(6): 20150074.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
22. Fernando C, Goldstein R, Szathmáry E: **The neuronal replicator hypothesis**. *Neural Comput*. 2010; 22(11): 2809–2857.  
[PubMed Abstract](#) | [Publisher Full Text](#)
23. Fernando C, Karishma KK, Szathmáry E: **Copying and evolution of neuronal topology**. *PLoS One*. 2008; 3(11): e3775.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
24. Fernando C, Szathmáry E: **Natural selection in the brain**. In: Glatzeder B, Goel V, von Müller A, editors. *Towards a theory of thinking. vol. 5 of On thinking*. Berlin Heidelberg: Springer-Verlag; 2010; 291–322.  
[Publisher Full Text](#)
25. Fernando C, Vasas V, Szathmáry E, *et al.*: **Evolvable neuronal paths: a novel basis for information and search in the brain**. *PLoS One*. 2011; 6(8): e23534.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
26. Kemp C, Tenenbaum JB: **The discovery of structural form**. *Proc Natl Acad Sci U S A*. 2008; 105(31): 10687–10692.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
27. Ullman TD, Goodman ND, Tenenbaum JB: **Theory learning as stochastic search in the language of thought**. *Cognitive Dev*. The Potential Contribution of Computational Modeling to the Study of Cognitive Development: When, and for What Topics? 2012; 27(4): 455–480.  
[Publisher Full Text](#)
28. Bórgers T, Sarin R: **Learning through reinforcement and replicator dynamics**. *J Econ Theory*. 1997; 77(1): 1–14.  
[Publisher Full Text](#)
29. Niekum S, Barto AG, Spector L: **Genetic programming for reward function search**. *IEEE Trans Auton Ment Dev*. 2010; 2(2): 83–90.  
[Publisher Full Text](#)
30. Sutton RS, Barto AG: **Introduction to reinforcement learning**. 1st ed. Cambridge, MA, USA: MIT Press; 1998.  
[Reference Source](#)
31. Harper M: **The replicator equation as an inference dynamic**. ArXiv e-prints. 2009.  
[Reference Source](#)
32. Shalizi CR: **Dynamics of Bayesian updating with dependent data and misspecified models**. *Electron J Statist*. 2009; 3: 1039–1074.  
[Publisher Full Text](#)
33. Campbell JO: **Universal Darwinism As a Process of Bayesian Inference**. *Front Syst Neurosci*. 2016; 10: 49.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
34. Hopfield JJ: **Neural networks and physical systems with emergent collective computational abilities**. *Proc Natl Acad Sci U S A*. 1982; 79(8): 2554–2558.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
35. Rolls ET, Treves A: **Neural networks and brain function**. Oxford, New York: Oxford University Press; 1998.  
[Publisher Full Text](#)
36. Carrillo-Reid L, Yang W, Bando Y, *et al.*: **Imprinting and recalling cortical ensembles**. *Science*. 2016; 353(6300): 691–694.  
[PubMed Abstract](#) | [Publisher Full Text](#)
37. Kilgard MP: **Harnessing plasticity to understand learning and treat disease**. *Trends Neurosci*. 2012; 35(12): 715–722.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
38. Watson RA, Jansen T: **A building-block royal road where crossover is provably essential**. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. GECCO '07. New York, NY, USA: ACM; 2007; 1452–1459.  
[Publisher Full Text](#)
39. Maynard Smith J: **The Evolution of Sex**. Cambridge University Press; 1978.  
[Reference Source](#)
40. Maynard Smith J: **The units of selection**. *Novartis Found Symp*. 1998; 213: 203–11; discussion 211–7.  
[PubMed Abstract](#)
41. Storkey AJ: **Efficient covariance matrix methods for Bayesian gaussian processes and Hopfield neural networks**. Imperial College, Department of Electrical Engineering, Neural System Group; 1999.  
[Reference Source](#)
42. Sompolsinsky H: **Computational neuroscience: beyond the local circuit**. *Curr Opin Neurobiol*. Theoretical and computational neuroscience. 2014; 25: xiii–xviii.  
[PubMed Abstract](#) | [Publisher Full Text](#)
43. Müller S, Appel B, Balke D, *et al.*: **Thirty-five years of research into ribozymes and nucleic acid catalysis: where do we stand today? [version 1; referees: 2 approved]**. *F1000Res*. 2016; 5: pii: F1000 Faculty Rev-1511.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
44. Mountcastle VB: **Modality and topographic properties of single neurons of cat's somatic sensory cortex**. *J Neurophysiol*. 1957; 20(4): 408–434.  
[PubMed Abstract](#)
45. Rolls ET: **Attractor networks**. *Wiley Interdiscip Rev Cogn Sci*. 2010; 1(1): 119–134.  
[PubMed Abstract](#) | [Publisher Full Text](#)
46. Soto D, Silvano J: **Reappraising the relationship between working memory and conscious awareness**. *Trends Cogn Sci*. 2014; 18(10): 520–525.  
[PubMed Abstract](#) | [Publisher Full Text](#)
47. Madl T, Baars BJ, Franklin S: **The timing of the cognitive cycle**. *PLoS One*. 2011; 6(4): e14803.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
48. Gustafsson B, Asztely F, Hanse E, *et al.*: **Onset Characteristics of Long-Term Potentiation in the Guinea-Pig Hippocampal CA1 Region in Vitro**. *Eur J Neurosci*. 1989; 1(4): 382–394.  
[PubMed Abstract](#) | [Publisher Full Text](#)
49. Hirsch JC, Crepel F: **Use-dependent changes in synaptic efficacy in rat prefrontal neurons in vitro**. *J Physiol*. 1990; 427(1): 31–49.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
50. Fernando C, Szathmáry E: **Chemical, neuronal, and linguistic replicators**. In Pigliucci M, Müller G, editors. Cambridge, MA: MIT Press; 2010; 209–250.  
[Publisher Full Text](#)



51. Nadel L, Land C: **Commentary-Reconsolidation: Memory traces revisited.** *Nat Rev Neurosci.* 2000; **1**(3): 209–212.  
[Publisher Full Text](#)
52. Nadel L, Moscovitch M: **Memory consolidation, retrograde amnesia and the hippocampal complex.** *Curr Opin Neurobiol.* 1997; **7**(2): 217–227.  
[PubMed Abstract](#) | [Publisher Full Text](#)
53. Stewart TC, Choo X, Eliasmith C: **Symbolic Reasoning in Spiking Neurons: A Model of the Cortex/Basal Ganglia/Thalamus Loop.** In: 32nd Annual Meeting of the Cognitive Science Society; 2010.  
[Reference Source](#)
54. Dehaene S, Kerszberg M, Changeux JP: **A neuronal model of a global workspace in effortful cognitive tasks.** *Proc Natl Acad Sci U S A.* 1998; **95**(24): 14529–14534.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
55. Shanahan M: **A spiking neuron model of cortical broadcast and competition.** *Conscious Cogn.* 2008; **17**(1): 288–303.  
[PubMed Abstract](#) | [Publisher Full Text](#)
56. Jacobs C, Silvanto J: **How is working memory content consciously experienced? The 'conscious copy' model of WM introspection.** *Neurosci Biobehav Rev.* 2015; **55**: 510–519.  
[PubMed Abstract](#) | [Publisher Full Text](#)
57. Kumar A, Rotter S, Aertsen A: **Spiking activity propagation in neuronal networks: reconciling different perspectives on neural coding.** *Nat Rev Neurosci.* 2010; **11**(9): 615–627.  
[PubMed Abstract](#) | [Publisher Full Text](#)
58. Hertz J, Palmer RG, Krogh AS: **Introduction to the Theory of Neural Computation.** 1st ed. Perseus Publishing; 1991.  
[Reference Source](#)
59. Oberauer K: **Access to information in working memory: exploring the focus of attention.** *J Exp Psychol Learn Mem Cogn.* 2002; **28**(3): 411–421.  
[PubMed Abstract](#) | [Publisher Full Text](#)
60. Fukai T, Tanaka S: **A simple neural network exhibiting selective activation of neuronal ensembles: from winner-take-all to winners-share-all.** *Neural Comput.* 1997; **9**(1): 77–97.  
[PubMed Abstract](#) | [Publisher Full Text](#)
61. Morita K, Jitsev J, Morrison A: **Corticoatrial circuit mechanisms of value-based action selection: Implementation of reinforcement learning algorithms and beyond.** *Behav Brain Res.* 2016; **311**: 110–121.  
[PubMed Abstract](#) | [Publisher Full Text](#)
62. Chklovskii DB, Mel BW, Svoboda K: **Cortical rewiring and information storage.** *Nature.* 2004; **431**(7010): 782–788.  
[PubMed Abstract](#) | [Publisher Full Text](#)
63. Holtmaat A, Svoboda K: **Experience-dependent structural synaptic plasticity in the mammalian brain.** *Nat Rev Neurosci.* 2009; **10**(9): 647–658.  
[PubMed Abstract](#) | [Publisher Full Text](#)
64. Churchill AW, Sigtia S, Fernando C: **Learning to generate genotypes with neural networks.** *Evolutionary Computation.* 2015.  
[Reference Source](#)
65. Maynard Smith J, Szathmáry E: **The major transitions in evolution.** Oxford: Freeman & Co. 1995.  
[Reference Source](#)
66. Reed A, Riley J, Carraway R, et al.: **Cortical map plasticity improves learning but is not necessary for improved performance.** *Neuron.* 2011; **70**(1): 121–131.  
[PubMed Abstract](#) | [Publisher Full Text](#)
67. Baluja S, Caruana R: **Removing the genetics from the standard genetic algorithm.** Morgan Kaufmann Publishers; 1995. 38–46.  
[Publisher Full Text](#)
68. Friston K, Buzsáki G: **The Functional Anatomy of Time: What and When in the Brain.** *Trends Cogn Sci.* 2016; **20**(7): 500–511.  
[PubMed Abstract](#) | [Publisher Full Text](#)
69. Zachar I: **IstvanZachar/Neurodynamics: Publication release.** 2016.  
[Data Source](#)